Amendments to the Specification

Please replace the paragraph on page 1, beginning at line 5 with the following paragraph:

The present application is related to Tsao, et al. U.S. Patent Application entitled "Systems and Methods of Routing Data to Facilitate Error Correction", Filed July 29, 2003, Application No. 10/632,206 ~~Attorney Docket No. 200312693-1~~, which is assigned to the same assignee as the present application and which is incorporated herein by reference.

Please replace the paragraph on page 1, beginning at line 14 with the following paragraph:

~~Error codes~~ Error control codes are commonly used in electronic systems to detect and/or correct data errors, such as transmission errors or storage errors. One common use of error control codes is to detect and correct errors with data stored in a memory of computer system. For example, error correction bits, ~~or check bits~~ also called check bits, can be generated for data prior to storing data to one or more memory devices. The ~~error or correction~~ check bits are appended to the data to provide a data structure that is stored in memory. When the data is read from the one or more memory devices, the check bits can be used to detect or correct errors within the data. Errors can be introduced, for example, either due to faulty components or noise in the computer system. Faulty components can include faulty memory devices or faulty data paths between the devices within the computer system, such as faulty pins.

Please replace the paragraph on page 1, beginning on line 25, with the following paragraph:

Error management techniques have been developed to mitigate the effects associated with these errors. One simple technique used for personal computers is known as parity checking. Parity checking utilizes a single bit associated with a piece of data to determine whether there is a single bit error in the data. Parity checking cannot detect multiple bit errors and ~~provided~~ provides no means for correcting errors. A more sophisticated system, such as a server, uses

error correction codes (ECCs) to detect and correct some errors. ~~An error correction code (ECC) consists of a group of bits, or codes, associated with a piece of data.~~ A typical ECC system may use eight ECC bits (check bits, correction bits) for a 64-bit piece of data. The ECC bits provide enough information for an ECC algorithm to detect and correct a single bit error, or to detect double bit errors.

Please replace the paragraph on page 2, beginning on line 1, with the following paragraph:

One error correction feature employed by servers is referred to in the industry as chipkill. The term chipkill refers to the ability to correct multiple bit errors in memory, where multiple bit errors are ~~based on~~ <u>confined within</u> the width of the memory device. For example, for a 32Mbit dynamic random access memory (DRAM) device that is 4 bits wide, a system that supports a chipkill function would be able to correct a 4-bit wide error in the memory device. Thus, the ~~failure of an entire DRAM chip during a DRAM cycle (*e.g.*, read operation, write operation) organized into a 4-bit width configuration~~ <u>failure of an entire 4-bit wide DRAM chip during a DRAM cycle (*e.g.*, read operation, write operation) in a configuration</u> that supports chipkill would not cause the system to fail. Chipkill allows a system to operate in the event of multiple bit errors in any one memory device.

Please replace the paragraph on page 2, beginning on line 19 with the following paragraph:

The present invention relates to systems and methods for detecting and correcting errors in data structures. The systems and methods employ error correction code (ECC) techniques that detect and correct errors in a data structure. The data structure is ~~partitioned into separate adjacent bit pair domains, such that a single adjacent bit pair from each memory device is assigned to a given domain~~ <u>partitioned into domains referred to herein as 'adjacent bit pair</u>

domains,' with each adjacent bit pair domain being assigned an adjacent bit pair from each memory device. Data associated with a given adjacent bit pair domain can include data bits and check bits that are employed by an ECC technique to detect and correct data bit errors (*e.g.*, single bit errors, adjacent double bit errors) associated with the adjacent bit pair domain.

Please replace the paragraph on page 3, beginning on line 23, with the following paragraph:

The present invention relates generally to systems and methods for detecting and correcting bit errors in data structures. The systems and methods employ error correction code (ECC) techniques that detect and correct single bit errors and adjacent bit pair errors in a data structure. The systems and methods are operative to ~~process data structures with more bits than can be detected and corrected by the ECC techniques employed~~ process data structures having more bits than can be detected and corrected by a single application of the ECC techniues employed. This is accomplished by ~~partitioning a data block and/or data structure into separate domains equal to the number of bits that can be processed by the ECC technique~~ partitioning the data structure (data block) into domains having a number of bits equal to the number of bits that can be processed by a single application the ECC techniques employed. Chipkill is ~~achieved~~ facilitated by populating the separate domains with adjacent bit pairs, such a single adjacent bit pair from each memory device is assigned to a given adjacent bit pair domain.

Please replace the paragraph on page 4, beginning on line 7, with the following paragraph:

FIG. 1 illustrates a system 10 for detecting and correcting errors in a data structure in accordance with an aspect of the present invention. The system 10 can be a server or other computer system that performs error correction associated with a system memory. The system 10 includes an error correction unit 12, a data separator/combiner 14 and a system memory 16. The system memory 16 is comprised of a plurality of memory devices 18, labeled memory device #1 through ~~#K, where K is an integer greater than one~~ #K, where #K is an integer greater

than one. The memory devices 18 can ~~be for example, but not limited to~~ be, for example but not limited to, single-in-line memory modules (SIMM), dual-in-line memory modules (DIMM) and dynamic random access memory (DRAM) modules or other types of memory devices. The system memory 16 is coupled to a data bus 24 and an address bus 26. The size of the data bus 24 and the data structure stored and read into memory in a single memory cycle is equal to the number of memory devices multiplied by the bit column width of the memory devices.

Please replace the paragraph on page 4, beginning on line 20, with the following paragraph:

For example, if the system memory is comprised of 72 memory devices having 4-bit ~~width columns~~ column width, then the size of the data bus 24 and data structure stored and read in a single memory cycle would be 288 bits. However, ~~an ECC checker and corrector for a 288 bit data structure would be impractical~~ an ECC checker and corrector for a 288-bit ECC codeword is considered impractical. Therefore, the present invention partitions the data structure into separate adjacent bit pair domains comprised of adjacent bit pairs from each memory ~~devices~~ device, such that a single adjacent bit pair from each of the memory devices is assigned to a given domain. Error detection and correction can then be performed on data bits associated with the separate adjacent bit pair domains ~~sequentially or in parallel~~ sequentially in less time than otherwise required, or in parallel.

Please replace the paragraph on page 5, beginning on line 6, with the following paragraph:

The error correction unit 12 includes an error corrector 13 and a check bit generator 15. The error correction unit 12 is operative to receive data blocks from the crossbar device, partition the data blocks into a plurality of adjacent bit pair domain data sets, and generate check bits for each of the plurality of adjacent bit pair domains. ~~The adjacent bit pair domains are populated with the check bits and data bits from the data block.~~ The check bits for the data block are then

added to the adjacent bit pair domains. ~~The adjacent bit pair domains are assigned adjacent data bit pairs per memory device as discussed above.~~ The adjacent bit pair domains are assigned to adjacent data bit pairs for each memory device. The number of adjacent bit pair domains is based on the column width of the system memory 16. For example, two adjacent bit pair domains are employed for memory devices with a 4-bit column width, while four adjacent bit pair domains would be employed for memory devices with an 8-bit column width.

Please replace the paragraph beginning on page 5, line 27, and ending on page 6, with the follow paragraph:

For example, during a read operation, a data structure associated with an address is read from the plurality of memory devices 18 and provided to the data separator/combiner 14 *via* the data bus 24. For example, if each memory device has a 4-bit column width, 4 bits from each memory device are provided to the data separator/combiner 14. If each memory device has an 8-bit column width, 8 bits from each memory device 18 are provided to the data separator/combiner 14. The data separator/combiner 14 partitions the data structure into adjacent bit pair domains. Each adjacent bit pair domain includes a data portion and a correction bit portion. The number of bits in an adjacent bit pair domain corresponds to the number of bits ~~correctable by the error corrector~~ correctable within a desired amount of time by the error corrector 13 in the error correction unit 12.

Please replace the paragraph on page 6, beginning on line 25, with the following paragraph:

It is to be appreciated that the error correction unit 12 can include additional correctors, such that correction can be performed in parallel on data bits associated with different domains to ~~facilitate speed associated with~~ facilitate faster error correction. Additionally, the error correction unit 12 can include ~~additional check bit~~ additional ECC codeword check bit

generators, such that check bits can be assigned and appended to data bits associated with different domains to ~~facilitate speed associated with~~ facilitate faster check bit generation.

Please replace the paragraph beginning on page 6, line 31, and ending on page 7, with the following paragraph:

FIG. 2 illustrates an adjacent bit pair domain configuration 40 in accordance with an aspect of the present invention. The adjacent bit pair domain configuration 40 ~~provides chipkill~~ facilitates chipkill functionality to system memory devices (*e.g.,* for servers) associated with ECC techniques that can correct single and adjacent double bit errors with memory data buses larger than the capabilities of ECC techniques. The domain configuration illustrates a plurality of memory devices 42, labeled memory device #1 through ~~K, where K is an integer greater than 1~~ #K, where #K is an integer greater than one. Each of the memory devices 42 has a row associated with a given row address illustrated in FIG. 2 as row address A. During reading and writing of row address A, data bits associated with the row corresponding to row A are concurrently provided at the system memory data bus. Each row of the memory devices 42 has a column width N, where N is an integer multiple of 4 (*e.g.,* 4, 8, 16, 32, etc.)

Please replace the paragraph beginning on page 7, line 33, and ending on page 8, with the following paragraph:

The system memory 66 is comprised of a plurality of DRAM devices 68, labeled memory device #1 through ~~#K, where K is an integer greater than one~~ #K, where #K is an integer greater than one. For illustrative purposes, the DRAM devices 68 of FIG. 3 will be discussed as being 4-bit column width devices. However, other column width devices (*e.g.,* 8, 16, 32, 64, etc.) can be employed in accordance with the present invention. The system memory 66 is coupled to a data bus 80 and an address bus 82. The size of the data bus 80 and data structure read and written to the system memory during a DRAM cycle is equal to the number of memory devices 68 multiplied by the bit column width of the memory devices 68. In the present example, the

system memory 66 is comprised of 72 DRAM devices with 4 bit-width columns. Therefore, the size of the data bus 80 and data structure stored and read in a single memory cycle is 288 bits.

Please replace the paragraph on page 11, beginning on line 1, with the following paragraph:

The system memory 126 is comprised of a plurality of memory devices 128, labeled memory device #1 through #K, where ~~K~~ #K is an integer greater than one. The system memory 126 is coupled to a data bus 146 and an address bus 148. The size of the data bus 146 and data structure is equal to the number of memory devices 128 multiplied by the bit column width of the memory devices 128. For example, the bit column width can be 4 with K being 72 to provide a 288-bit data structure. The data separator device 124 is coupled to the system memory 126 *via* the data bus 146. The memory controller 122 is coupled to a crossbar device (not shown) over a crossbar bus 142 and a mid-bus 144. The crossbar device can be coupled to a plurality of microprocessor devices, input/output devices and/or memory devices (*e.g.,* one or more cache memory devices). The mid-bus 144 connects the memory controller 122 to the data separator device 124.